Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 9, No. 4, 2021, 525-541.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

An Efficient Hybrid Method for Semantic Web Service Discovery

Pourya Farzi and Reza Akbari*

Department of Computer Engineering and Information Technology, Shiraz University of Technology, Shiraz, Iran.

Article Info

Article History:
Received 10 August 2020
Revised 25 June 2021
Accepted 07 October 2021

DOI:10.22044/JADM.2021.9958.2132

Keywords:
Semantic Web Service, Matchmaking, Classification, Latent Semantic Analysis, Word Semantics.

**Corresponding author: akbari@sutech.ac.ir(R. Akbri).*

Abstract

The web service is a technology for defining the self-describing objects, and the structural-based and loosely coupled applications. They are accessible all over the web, and provide a flexible platform. Although the service registries such as Universal Description, Discovery and Integration (UDDI) provide facilities for the users in oder to search for the requirements, retrieving the exact results that satisfy the users' needs is still a difficult task since the providers and requesters have various views about the descriptions with different explanations. Consequently, one of the most challenging obstacles involved in the discovery task would be how to understand both sides, which is called the knowledge-based understanding. This is of immense value for the search engines, information retrieval tasks, and even NLP-based various tasks. The goal is to help recognize the matching degrees precisely and retrieve the most relevant services more straightforward. In this research work, we introduce a conceptual similarity method as a new way that facilitates the discovery procedure with a less dependency on the provider and the user descriptions to reduce the manual intervention of both sides and is more explicit for the machines. We provide a comprehensive knowledge-based approach by applying the Latent Semantic Analysis (LSA) model to the ontology scheme-WordNet and domain-specific in-sense context-based similarity algorithm. The evaluation of our similarity method, done on the OWL-S test collection, shows that a sense-context similarity algorithm can boost the disambiguation procedure of descriptions, which leads to a conceptual clarity. The proposed method improves the performance of service discovery in comparison with the novel keyword-based and semantic-based methods.

1. Introduction

Service-oriented architecture is known as a modern technology for software development. Recently, a vast majority of software systems have been planned by the web services. These entities are seen as activities in the business community. They often lead to intangible benefits or outcomes. Moreover, the terms "web service" and "service" are frequently regarded as replaceable to name an achievable software entity in the computer science. The providers of services register and advertise them within UDDI [1] registry by providing some details such as service provider information, pointers, and descriptions. This data defines a model and API for service registry used in the discovery, selection, and composition process done by the users or agents. However, the agents, providers, and end-users may have a various understanding of each service in terms of definition and description, which is why this topic has provoked controversial issues.

Considering the recent web progress (web 1.0 and web 2.0), there seem to be some obstacles in the way of its progress. One of them is associated with the information retrieval scope that consists

primarily of the keywords on web 1.0 and web 2.0. Consequently, it gives rise to a barrier to web achieving a high accuracy regarding the user search results [2]. The process of finding the appropriate web services that fulfill unique requirements is called 'discovery', the way of which is among the most critical challenging aspect of a service scheme. Certainly, the web services will be useless if they cannot be discovered. This occurs due to the vagueness of the written natural language. As a case in point, the documents containing synonyms or related words of the query keywords might not be retrieved by search engines utilizing naive keyword-based, even though they may be relevant and must be included in the retrieved documents. There is another problem, namely homonyms. Take searching "mouse", for example, the search engine will return the pages containing the data about both "mouse" (a part of computer system) and "mouse" (an animal) to the user who is interested in only one of them.

Despite all the remarkable efforts that have been made in the service discovery engines, there are several shortcomings. A variety of systems and approaches have been developed and extended to the search web services lately, for instance, WADL (web application Description Language), WSDL (web service Description Language) or rest APIs that do not offer any formal semantics [3]. In order to more accurately match, it is vital to acquire their underlying semantics. Next, other techniques have employed the classic information retrieval strategies for determining the similarity among two descriptions of web services without consideration for domains [4]. Although these techniques attained a noticeable accuracy for retrieving proper services, comprehensive solution concerning homonyms and complexity of understanding the natural language for systems remains an open-ended question.

In order to deal with the above problems, we have designed a method to ease the challenges of suitable web service discovery, which contains a two-step matching process. The first step reduces the search space to sub-spaces since the larger the service pool is, the more time-consuming it would be for the engines or systems to fulfill the requirements of the requesters due to more calculations. In the second step, more importantly, we compare a query with the services belonging to these smaller sub-spaces. The focus is on the matchmaking part comprised of word sense disambiguation and sense context similarity. By proposing a context-aware algorithm, we model the discovery process with the help of NLP and semantic approach. Before the main steps, there exists a pre-processing phase discussed in details later. In addition, an LSA model [5] and a thesaurus are used for adding semantic to the model in order to facilitate a homonymous interpretation. LSA is a method applied in natural language processing, which helps to analyze the relationships between a set of documents and the terms they contain. We exploit it as a powerful means in the application of information retrieval and finding the word-word similarity [6]. In terms of the evaluation presented in Section IV, our technique makes the discovery aim more efficient in comparison with the previous methods respecting information retrieval measurements such as F-score. Further details are explained in that section. The proposed method has the potential to deal with a large number of services.

The main contributions of this paper are as what follow. 1) A new pipeline is designed to pre-process the provider and requester descriptions, which, in turn, extract the helpful and required data by various modules since it helps to ignore the unnecessary data and to enrich the descriptions. 2) A sense context-based similarity algorithm is designed and implemented that has the benefit of a deeper layer of senses by the application of the related word senses of every extracted sense. 3) A concept similarity formula comprising a combination of the LSA model and two similarity metrics is provided.

The remainder of this paper is prepared as what follows. The subsequent part surveys the most relevant state-of-art works; furthermore, a taxonomy of some related works is given. Section III shows the proposed method in detail, and then the experimental measurements are presented in Section IV. Finally, the last section ends with a conclusion and the future works.

2. Related Work

The web service discovery has become a sufficiently challenging subject in the recent years as a result of the development of service-oriented architecture and computing, and the researchers have provided various kinds of approaches, one of which is a combination of the web service technologies and the semantic web [7]. For example, WSDL-S, as a mechanism, appends ontologies into the service descriptions written in WSDL [8]. Another example is web service modeling ontology (WSMO), which makes a full use of a perceptual framework and a formal language to define the related facts of services semantically. In practice, this solution permits the developers to simplify the automation of

invoking, discovering, and combining services over the internet. It is transparently obvious that this framework has been established based on an ontology scheme [9], with which the service concepts can be associated. The chief advantage of defining the services by ontologies is to access an unambiguous definition of the service elements such as inputs and outputs. Consequently, the software agents can efficiently function in the discovery, selection, and composition of the services [10]. Nevertheless, creating and extending ontologies have been known as a costly, time-consuming, and tedious function [11]. Furthermore, the efficiency is reduced due to the complexity, reusability, integration, and lack of semantically annotated services.

Lately, the information retrieval techniques have been proposed in order to resolve the mentioned problems. One of the marked models is the Vector Space Model (VSM) made up of vectors, each of which represents a description [12]. In other words, every vector has several dimensions and each dimension indicates a separate term. In addition, the term that occurs in the document is equal to a non-zero value vector, and several ways have been developed to compute these values [12].

Paolucci *et al*. have implemented an algorithm that proposes matchmaking by a greedy approach. It uses the IO concepts in order to determine four degrees of the match [13].

Generally, the semantic service discovery approaches attempt to perform similarity matchmaking based on semantic annotation and ontology [14]. Semantic matching can overcome some deficiencies of syntactic discovery but they are more intricate than the keyword-based methods [14].

We categorized various types of semantic service discovery models. Although many aspects could be considered, our categorizing scheme is made up of three classes, namely the domain ontology-based, public ontology-based, and semantic and syntax-based approaches [15].

Domain ontology-based: All the methods in [16]–[23] focus on the domain ontology-based approach. Oleshchuk *et al*. [16] have provided a semantic similarity system for extracting knowledge from the ontological patterns. Paliwal *et al*. [17] have focused on service categorization and selection by the semantic-based techniques. Parameter-based service refinement and enhancing the user request were the basic components of their approach that required mapping. The ontology mapping process is one of the critical challenge because it requires many calculations of the similarity of entities extracted from heterogeneous ontologies [18].

Two preceding methods have been explained in a survey performed by Pakari *et al*. [19]. They divided web service discovery into three aspects: architectural, automated, and matchmaking view; their experiments were particularly conducted on the semantic-based approaches. Nevertheless, our focus is on the matchmaking view that is divided into semantic-based, syntax-based, and context-aware.

In 2008, a prototype was planned for a uniform graph-based ontology representation and service descriptions. It integrated the structural and syntactic matching of graph representations by the isomorphic properties [20]. Furthermore, a conceptual design has been implemented by Zhou [21] premised on three components: vector space model matching, hierarchical ontology matching, and quality of service matching. In their three-layer method, the fundamental layer is the semantic distance implemented by the vector space model. Next, the web services are matched with ontology hierarchically in the second layer; Lastly, they are matched by the QoS features.

Bener *et al*. [22] have proposed the matchmaking condition expressions in the OWL-S files written in the SWRL format. They succeeded in detecting fairly precise pairing among conditions, arguments, and predicates belonging to the request and advertisement. Their method was modified by considering the bipartite graph matching. They calculated the bipartite graph weights on a comprehensive scoring mechanism that included WordNet component, sub-sumption, and semantic distance. Changbo Ke *et al*. [23] have made a complete use of ontology trees for elucidation of the requirement document and service profile. Attribute, structure conception, and similarity of corresponding trees nodes were calculated through the hierarchical and taxonomic methodology. Afterward, a whole series of constraints were defined as a connection between the structures and the conceptions to attain the restructuring regulations. Some advantages and disadvantages are investigated here. First, these methods reduce the manual web service discovery, the major parts of which could automatically be conducted by the software agents, namely identification, integration, and execution. Secondly, the number of similar services, as an indirect factor, has influenced the final accuracy of service discovery. This factor could be improved by enhancing the semantics formats. However, either the end-users or the agents should expand their basic knowledge of the semantic web services. In this case, the

problem becomes even more complicated when it comes to the situation that the intermediate agents or requesters may not be informed of all the domain ontology knowledge. One more limitation seems to be a semantic matching mechanism because the most recent researchers have presumed that both the requester and the provider take the same ontology domain to service description; nonetheless, it is not applicable in the real-world scenarios. The ontology mapping techniques are required due to triumphing over heterogeneity, coordinating the diversity of ontologies to support interoperability [24]. In the other method, a hybrid semantic service discovery module has been introduced by M. DEEPA *et al*. [25], who have offered semantic discovery-based totally on both the non-functional and functional properties of the OWL services. They used the bipartite graph-based approach for matching the service parameters, and the natural language processing techniques were utilized as a method for the textual-description of a web service in the non-functional part.

Public ontology-based: In [26]–[28], the researchers have explained some methods based on the public ontology. In [26], a minimal model has been presented for information processing of the RESTFUL services, where WordNet is applied as a public ontology to enhance the semantic features of services. Additionally, various similarity metrics were employed in the model. In a sense, they achieved a high discovery precision, while avoiding the cost of the ontology-based methods. A two-step semantic filtering method was offered to identify appropriate services with reliable score calculations during the discovery phase [27]. The assignment idea was combined with the service framework to discover the web services automatically and briskly, which means the matchmaking procedure is transformed into the assignment process [28]. In [29], a matching service clustering has been presented for the discovery aim, which utilizes a new semantic measure based on the functional and process hierarchical similarity. They made use of the principal metrics of the similarity mentioned in [30]. Another approach has been published by Pushba *et al*. [31], who have introduced OntoDisco as a semantic aware method. A word-level semantic was applied using clustering premised on the NPMI value. An ontology for the principal classes of web services was designed to improve clustering, which used the initial aggregation of the data and the concepts.

All the above-mentioned studies have some benefits and drawbacks, and we state the key ones. The first advantage is to allow the developers to enhance the semantic information of web services without considering semantic annotation or ontology. Next, using the thesauruses such as WordNet, which is not domain-specific, avoids plenty of web service complications [26]. However, WordNet does not support the multi-lingual approaches that are located in a fine-grain category.

Semantic and syntax-based approaches: The Syntax-based and semantic-based methods have been presented in [32]–[35]. C. B. Merla [32] has proposed the information structures that have been obtained from the semantic user context data at the design time. Klusch *et al*. in [33] have presented a hybrid matching based on the experimental results of OWLS-MX that has improved logic-based matching for the semantic services. The operation-based search with bipartite graph algorithm has been discussed in [34] in order to ease the discovery process. Also Farooq *et al*. have described an effective hybrid technique by merging the synonym-based search and the input-parameter search [35]. They attempted to propose a simple-term syntax in a keyword list that not only allowed the users to search by keywords and phrases but led to simplicity and clarity. Kokash *et al*. in [36] have proposed an algorithm concerning the interface similarity measurement for WSDL with the application of vector space model, wordnet, and semantic similarity metrics.

One limitation in the service discovery scope is just relying on the keyword search. In other words, the user cannot describe the search request more accurately, for the end-users should have a particular knowledge of the semantic web services, or else it would become difficult for them to retrieve the purposed services. As a consequence, these methods are not suited for automatic processing due to the lack of tackling this key issue well. Other presented approaches using semantic approaches to cover this issue suppose that the requester and provider apply the same ontology for the domains, which would not be workable in many cases. Therefore, we propose a method considering all this information. The reason for why we apply lexical semantics is that the meaning of a word is fully reflected by its context. A taxonomy of some of the related works along with the proposed method are in Table 1.

3. Proposed Method

The steps of the proposed method have been described in detail, which are based on the lessons learned from the previous taxonomy. It is

composed of classification and matching the semantic service information extracted from the description model. The general scheme shown in Figure 1 has four phases.

First, both services and queries (query and request are equal) have been pre-processed according to five steps, which are mentioned in the following section. This is done to removing the unnecessary data from the descriptions of services and queries. In the next phase, the pre-processed query has been received by the word sense disambiguator module for extracting the meaningful information and finding the concept similarity by context-based senses with a similarity formula. The services, then, have been classified into similar groups based on their functionality in the third phase, which helps the engines and agents to perform less calculation in terms of the number of services by focusing on a smaller number of services.

Afterward, the semantic similarity matchmaking task has been performed in two steps in the final phase: class and web service matching. The reason for the paper is discovery, accomplished by these two steps. Indeed, classification is a key step forward to assist the discovery of web services.

Table 1. A taxonomy of the related works along with the proposed methods

Approaches	**Matching object**	**Semantic annotation**	**Matching type**	**Context awareness**	**Layer-based**	**Classifier algorithm**	**Type of similarity measurement**	**Domain ontology-based approach**	**Public ontology-based approach**	**Logic non-logic**
Proposed method	I/O + Description	OWL-S	Hybrid	✓	✓	Naïve Bayes	LSA Model+ Leacock & Chodorow+ Jiang & Conrath	✓	✓	Hybrid
F.chen *etal*. [29]	I/O + process	OWL-S	Semantic			DB-Scan	LIN[30] + oppositeness	-	✓	Hybrid
Kokash[36]	Interfaces	WDSL	Syntactic	-	-	-	Lexical similarity	-	✓	Non-logic
Paolucci *et al*., 2002[13]	I/O	DAML-S	Semantic	-	-	-	Lexical similarity	✓	-	logic
M. Deepa *et al*. [25]	IOPE+ description	OWL-S	Semantic	-	-	-	Bipartite graph-based approach+ NLP	✓	-	Hybrid
Matthias Klusch *et al*. [33]	I/O	OWL-S	Hybrid	-	-	-	Hybrid measurement	✓	-	Hybrid
A.V. Paliwal et *al*. [17]	I/O + Description	WSDL	Semantic	-	-	SVD	LSI	✓	-	Non-logic
R. Karimpour *et al*. [26]	I/O	WSDL	Semantic	-	-	-	matching utility	-	✓	Non-logic
OntoDisco[31]	Interfaces	OWL-S	Semantic	-	-	K-means clustering	data reduction	✓	-	Non-logic

The OWL-S test collection is selected as the main repository in this work. Through the repository, the providers have published advertisements according to a specific description model. Initially, the services are crawled across the repository. Next, the Naïve Bayes algorithm is employed to arrange the services in classes. Then we conduct a matching process by the search module to distinguish the most appropriate services. This procedure is done in two steps: 1) the query is compared with the class labels, and consequently, the most relevant label is selected for the next step. 2) the web services are arranged in a rank by the Semantic Similarity Measure specified in (7). By classification, the matchmaking algorithm calculates the similarity between the request and a severely limited number of services, which means that the matchmaking has not undergone a complex process owing to copious numbers of services. Thus the time and computational cost would decline when having a smaller number of services to be handled. Further details will be thoroughly presented in the following sub-section.

We consider each request to match with the services in the matchmaking phase. A sample of service profile could be similar to Figure 2; in the economy domain, it is a service that provides the camera price and the company name. In a greater detail, the profile is composed of two inputs, two

outputs, and an atomic process. Turning to Figure 3, the service request has the same format as the advertisement does.

3.1. Pre-processing Phase

A brief description of the OWL-S service should be presented before the explanation of the way we propose to deal with the problem. Thus OWLS-TC3 is the third version of the service retrieval dataset. This test collection consist of five sub-directories, the first and second one are the service and query, respectively, gathered to support the performance evaluation of the service matchmaking approaches. It provides 1007 semantic services written in the OWL-S form, versions 1.1 and 1.0, from seven different categories (food, medical, travel, education, communication, weapons, economy). Furthermore, Ontology, domain, and WSDL are the other sub-directories. In addition, 29 test queries are embedded for the evaluation experiments [37].

OWL-S provides the comprehensive data of all the required elements in the form of XML tags applied to various types of domains, which is why we decided to extract only the required information to minimize the overhead on extensive data in few phases. Initially, the service descriptions and queries are extracted from the dataset used as the inputs of the method, which starts by two separate branches. The first branch illustrates the five steps of the user queries' pre-processing. The other one demonstrates the services' pre-processing phase. Although each module of every pipeline is proposed in different fields and somehow in the task at hand as well, these two branches show unique pipelines that are applied in this sequence for the first time, as depicted in Figure 1.

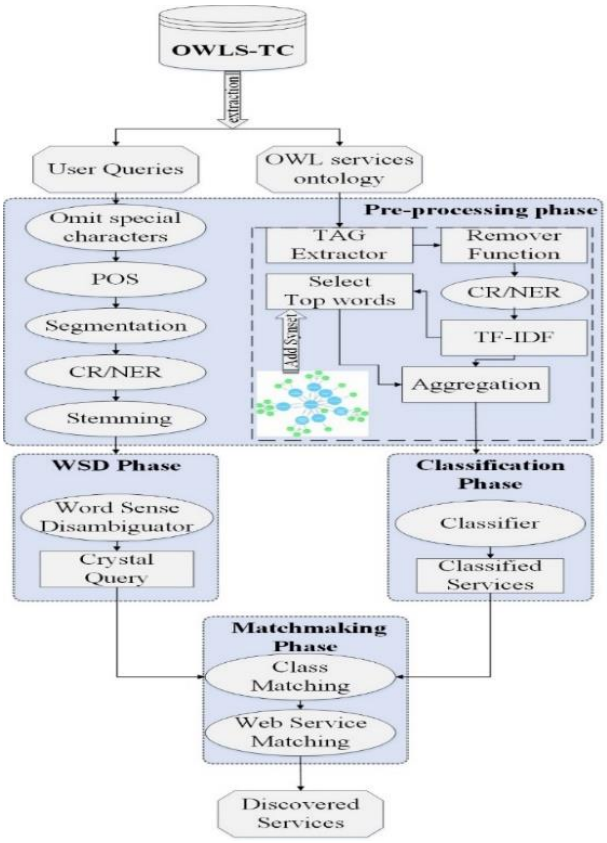

Figure 1. General scheme for the proposed method.

```
<profile:Profile rdf:ID="_PRICECAMERA_PROFILE">
<service:isPresentedBy rdf:resource="#_PRICECAMERA_SERVICE"/>
<profile:serviceName xml:lang="en">
WallmartCPriceService
</profile:serviceName>
<profile:textDescription xml:lang="en">
This service returns list of camera models and their prices available in Wallmart.
</profile:textDescription>
<profile:hasOutput rdf:resource="#_PRICE"/>
<profile:hasOutput rdf:resource="#_CAMERA"/>
<profile:has_process rdf:resource="_PRICECAMERA_PROCESS" /></profile:Profile>
```

Figure 2. Advertisement of a camera price service.

```
<profile:Profile rdf:ID="_PRICECAMERA_PROFILE">
<service:isPresentedBy rdf:resource="#_PRICECAMERA_SERVICE"/>
<profile:serviceName xml:lang="en">
WallmartCPriceService
</profile:serviceName>
<profile:textDescription xml:lang="en">
This service returns list of camera models and their prices available in Wallmart.
</profile:textDescription>
<profile:hasOutput rdf:resource="#_PRICE"/>
<profile:hasOutput rdf:resource="#_CAMERA"/>
<profile:has_process rdf:resource="_PRICECAMERA_PROCESS" /></profile:Profile>
```

Figure 3. A request for camera price service.

Omitting special characters: a query mainly consists of different elements such as the stop words and punctuations that do not assume a significant role by any means. Consequently, two sub-steps help purify the data as follows–removing the stop words and punctuations.

Removing stop-words: all tokens are compared with a list of stop-words that have been collected previously. It should be noted that the tag names, IP addresses, and some pieces of information are considered as the stop-words because not only are they identical in all of the queries but they do not convey any specific meaning and may also reduce the accuracy of the classification algorithm. The improved porter stemmer is deployed for this purpose [38]. Eventually, the outcomes are stored in an array set.

Removing punctuations: the sentences' delimiters are common signs in the dataset; hence, one of the refinement stages is then being ignored owing to the same reason as the stop words are removed, i.e. the punctuation marks are identified and omitted by the punctuation module.

Part of speech tagging: Concerning the role of the words in sentences, the POS tagger module is also applied in order to extract four kinds of parts of speech–i.e. the nouns, verbs, adjectives, and adverbs [39].

Segmentation: This process is required to identify the idioms and phrasal verbs such as "fill in" and "give a hand". If these idioms and verbs are not distinguished precisely or are considered as separate vocabulary items, they will not articulate the real meaning because "fill in" means "complete" or "inform", and also the second item conveys helping. About this, Wiktionary is searched for with the n-grams method, and the outcome of the above example could be "fill-in" and "give-a-hand". In Figure 3, the shopping mall is considered as a collocation in the segmentation procedure. In addition, some separate words might be stuck to each other owing to the human (or machine) error, and in some cases, this may deliberately happen such as the definition of the service name in OWLS-TC. As it can be observed in Figure 3, the service name, ShoppingMallCameraPriceService, should be processed in this way. As a result, each word is separated from the next one; therefore, the outcome of the process can be represented in this form-"Shopping Mall Camera Price Service"

CR/NER: This step involves two tasks, the first of which is co-reference resolution (CR). If some expressions in a specific text refer to the same element (thing or person), then they should be tagged as unique entities. In addition, with the help of named entity recognition (NER), which enhances the recognition of those entities with the same meaning, this procedure can be boosted [40]. Take E-book as an instance in the context of "education". This name entity can be introduced by the electronic book or even in the different forms of aliases such as "PDF" and "Portable Document Format". If identified correctly, we can improve the accuracy of the co-reference resolution model in Stanford CoreNLP. The following two sentences S1, S2 are cases in point. The extracted co-reference chains for service name and description {"Shopping Mall Camera Price Service", "This"}, {"analog camera", "their"}, {"digital camera", "their"} are presented, and each pair is tagged by a unique identification.

S_1: Shopping Mall Camera Price Service

S_2: This service returns a list of analog and digital camera brands/types and their prices available in a given shopping mall

Stemming: In the information retrieval and linguistic morphology, stemming is the process of reducing the inflected words to base form (root), which is of major benefit to grouping words and feature reduction. We have implemented it on an array-set mentioned previously.

To sum up, the result of example would be presented as follows:

S_1': W1

S_2': W1 return~~s~~ list of W2 brand~~s~~ and W3 type~~s~~ and W2, W3 price~~s~~ available in a given shopping-mall

On the other hand, the required information for service classification is elicited from the descriptions. Indeed, the main purpose is to extract the useful data in each description. Thus the services' pre-processing phase consisting of five steps is accomplished as what follows.

Tag extractor: concerning each service description, it comprises the XML (Extensible Markup Language) tags that lead us to apply an XML parser to read and extract each tag separately as regards preparing data for the next step.

Removing function: Two sub-steps of user queries' pre-processing, removing stop-words and punctuations, are considered, in a similar manner as above, to extract the service name, description, inputs and outputs, and ontology name. The purification of the data helps to facilitate the readability and further computational overhead to a large extent.

CR/NER: Since the structures of both queries and services are the same, this step is carried out similar to that previously discussed in pre-processing of the queries.

S3: Wallmart C Price Service

S4: This service returns the list of camera models and their prices available in Walmart.
The output is as follows:
S_3': W4
S_4': W4 returns the list of W5 models and W5 prices available in Wallmart

TF-IDF: There is no doubt that each of the elements in the service information does not have an equal importance. Thus we used a statistic method, called Term Frequency-Inverse Document Frequency, in order to distinguish between the words. Almost 83% of the recommender systems according to the digital library have employed this strategy.
This step comprises two parts; TF (t,d) illustrates term frequency, which signifies the number of times that word t occurs in document d. Following this, IDF (t, D) is the logarithm of the inverse fraction of documents that include the word t.

Selecting top words: All the words are then ranked concerning the values obtained by the previous step, i.e. this step makes the proposed method highlight the words with the highest value of TF-IDF as the top words. Next, the related synsets and synonyms of the top words are added to the data. This procedure is performed for each top word in every service. Marking the keywords and adding semantic meaning are the main purposes of doing the stage with the help of RITA- a java-based WordNet interface, i.e. WordNet, as a public ontology, helps to make every word more clear.

Aggregation: At the end of the process, all the main data from two previous steps is collected in a file according to each service, thereby, showing the abstract presentation of a service S=<I/O, S.D, TW>.
By performing the above steps, service name, service description, input and output, process names, WSDL file location, and ontology names are extracted from the service description. In the next stage, the Naïve Bayes algorithm by using the extracted features is used to categorize the services in the same domains as the test collection is categorized. The purpose of classification is that the matching algorithm just operates on a limited domain due to time-saving efficiency.
According to the mentioned steps, the outcome of the advertisement and request in Figures 2 and 3 are presented in Table 2.
Now, the WSD module (word sense disambiguation module) is equipped with domain and public ontology, and the LSA module receives the extracted data from the preceding phase.

3.2. Word sense disambiguation (WSD) phase

The required data for this phase is the refined queries that are received from the first phase. Indeed, this phase is responsible for finding similar concepts based on the word-to-word similarity metrics.
In the recent years, a great number of word-based similarity metrics have been developed [41]–[43]. The first class of these metrics includes knowledge-based metrics, where the semantic similarity between the words is estimated according to dictionaries or thesaurus, the most distinct advantage of which is that they can be very reliable because of them being experts' judgments; however, having said that, this way of matching requires a basic precondition regarding the class of words and part of speech that should be identical in both sides of matching.

Table 2. output of the first phase for the given example.

parameters	Service advertisement information	Service request information
Service name	Wallmart camera price service	Shopping mall camera price service
Description:	W4 return list of W5 model and W5 prices available in Wallmart	W1 returns list of W2 brands and W3 type and W2,W3 price available in a given shopping-mall
Inputs:	-	Shopping Mall
Outputs:	Camera, price	Camera, price
Domain Ontology:	Extended camera	Mid-level-ontology, extendedCamera

These schemes can compute the similarity between the contents of the same type, for instance, between one verb and the other [44]. Finding the correct meaning of the words is called word sense disambiguation, and is still an arduous task in NLP, which is why we implemented the WSD module in the proposed method. The second class of word-to-word similarity metrics relies on representations of the word meanings. In this class, the meanings of the words are represented as vectors into a high multi-dimensional space, where each dimension represents a latent semantic concept [44].
Our proposed method takes advantage of the above class as well. We expected that these features should lead to a reasonably comprehensive model. In addition, the following word similarity metrics were employed in the LSA model: Leacock & Chodorow [41] and Jiang & Conrath [45]. For the most part, the required steps to do so are detailed in the following sub-section.
The services reflecting the operation sets are modular. For this point, we referred to them as the

operations. According to what was said, each service could be shown as a 6-tuple :$(S_n, S_d, S_a, S_c, S_i, S_o)$, where the items indicate the name of service, description, concept to annotation for each service, context set of service, service input, and output, respectively. Similarly, the service requirement is presented by 6-tuple $R = (R_n, R_a, R_c, R_i, R_o, \Theta)$, where Θ is considered as the threshold of service requirement and the other tuples equivalent to service's tuples. The other critical parameter $S_i = \{ia_1, ia_2, \dots, ia_n\}$ is the service inputs, where $ia_j = (i_j, a_j)$ shows that the annotation of i_j is a_j, $1 \leq j \leq n$; likewise, $S_o = \{oa_1, oa_2, \dots, oa_n\}$ is the outputs of the service, where $oa_j = (o_j, a_j)$, the annotation of o_j is a_j, $1 \leq j \leq n$.

The word sense disambiguation module (briefly called WSD) is proposed for assigning an ambiguous word to the related sense by context. For example, a mouse has three different senses in Word-Net ontology: 1) an animal, 2) a part of a computer, 3) searching. Thus when the mentioned word is used to service annotation, the right sense is returned by context. The process of WSD is shown in Figure 4.

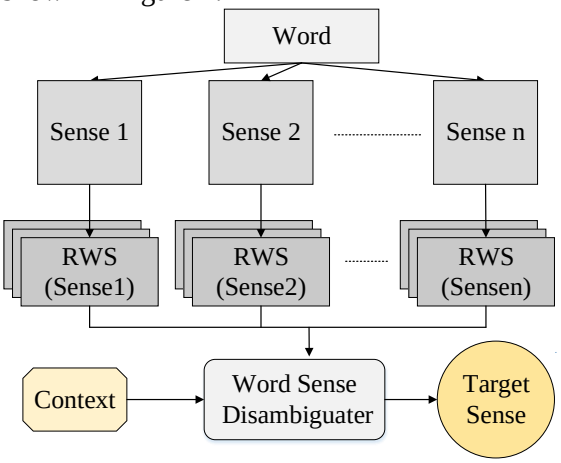

Figure 4. Word sense disambiguater.

In our example, "camera" is considered as the context parameter in the WSD algorithm. The context is the main parameter to limit the model to a specific domain.

As it could be observed in Figure 5, the first algorithm (service matching) is used to decide whether the service matches the service requirement or not; furthermore, it has been designed based on two algorithms: ServiceSimilarity and ServiceInterfaceSim. The service matching pseudo-code is a modified version of that presented in [46]. Indeed, the similarity degree of the service annotation is computed through the ServiceSimilarity algorithm presented in [46]. If this value is less than Θs, it will return false. This procedure is done the same as above for the procurement interface similarity.

The ServiceInterfaceSim algorithm is applied in order to calculate the annotation similarity between the requirement and the services; moreover, two constraints should be satisfied for checking the inputs and outputs:

| outputs of service | >= | outputs of requirement |

| inputs of requirement | >= | inputs of service |

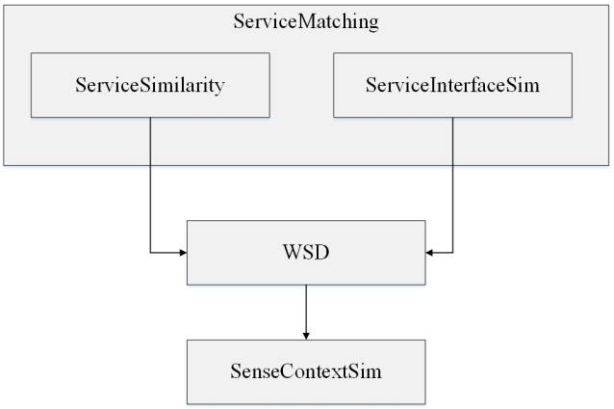

Figure 5. A flowchart of the proposed algorithms.

The service similarity and service interface similarity algorithms are precisely designed for matchmaking, and are applied in the WSD algorithm. First, each sense of the word is verified by this module, and it is presumed to be the right sense. Next, it is compared with the context by SCS. Then the next word is selected and compared in the same steps as the previous word. Finally, if the latter score was higher than the former one, then it would be selected as the best sense (steps 2-5 in Figure 6).

```
Input: Context (C), Word (W).
Output: bestSense
1. max_similarity;
2. for (each Sen of W){
3.       if (max_similarity < SCS(Sen,C)){
4.            max_similarity = SCS(Sen,C);
5.            bestSense = Sen;}}
6. Return bestSense;
```

Figure 6. WSD algorithm.

We designed and implemented the SenseContextSim (SCS) algorithm depicted in Figure 7 for computing the similarity values between Sense and Context. It shows the context-aware concept similarity, through which the related word senses are considered to help achieve the goal of discovery.

For the SenseContextSim algorithm, a new variable is defined as the output result in step (1). The related word senses for each sense and each context are stored based on *SenWS* and *ConWS* in the steps (2) and (3), respectively. If the *ConWS*

size was smaller than *SenWS* size, it would be selected for finding an optimal match between *ConWS* and *SenWS*. Next, the optimal word-to-word alignment in step (4) is applied to the job assignment problem (or assignment problem) defined as one of the fundamental combinatorial problems in the optimization field, and it is responsible for finding a maximum weight matching in a bipartite graph [47].

In the present method, the lowest part of the semantic similarity is modeled as the semantic word similarity. Afterward, a maximum weight matching is applied to discover the highest relations between ConWS and SenWS. This way of matching includes a fitness function designed according to the word-to-word semantic similarity metric, as shown in (1). In the following formula, "c" shows the number of permutations:

$$\sum_{C \in ConWS} concept\ similarity\ (C, map(ConWS \rightarrow \text{SenWS})) \quad (1)$$

concept similarity function is in charge of word to word similarity; additionally, *ConWS* and *SenWS* define the words from two different texts such as T1 and T2, respectively.

At the end of the process, there are the best word senses, each of which is extracted from the query by the WSD module. In other words, each query is expanded based on the context, and the crystal query is generated as a desirable outcome.

Input: Sen, Con
Output: SimilaritySC
1. SimilaritySC = 0;
2. SenWS=RelatedWordSense (Sen);
3. ConWS=RelatedWordSense(Con);
4. If (|SenWS|>|ConWS|)
   { Find an appropriate map: ConWS → SenWS, by considering optimal word-to-word alignment which maximize
   $\sum_{C \in ConWS} concept\ similarity\ (C, map(ConWS \rightarrow \text{SenWS}))$
   Return
   $\text{SimilaritySC} = \sum_{C \in ConWS} concept\ similarity\ (C, map(ConWS \rightarrow \text{SenWS}))/|ConWS|$ };
5. Else { find a map: SenWS → ConWS, by considering optimal word-to-word alignment which maximize
   $\sum_{C \in SenWS} concept\ similarity\ (C, map(ConWS \rightarrow \text{SenWS}))$
   Return
   $\text{SimilaritySC} = \sum_{C \in SenWS} concept\ similarity\ (C, map(ConWS \rightarrow \text{SenWS}))/|\text{SenWS}|$ };

return SimilaritySC

Figure 7. SenseContextSim algorithm.

Take the case of a service advertisement defined by {(car price service), (find, company, offer, same, car, range, given, amount, money), (automobile, cost, corporation), (Automobile, price), (car, amount of money), (price, company) } and a request {(car price service), (return, price, car), (automobile, cost), (Automobile, price), (car), (price), (0.75)} for example.

The first step is to check all the input words to figure the target sense out, which is matched with context to find the best similarity score. In this case, 'car' has three various senses: automobile, wagon, and vehicle. The nearest (highest similarity) sense is automobile based on the context set. Moreover, the other words undergo a similar process for both service requests and advertisement.

3.3. Service Classification

The classification is defined as the methodical grouping of objects into classes on the foundation of the structural association between them. In this step, the classification algorithm is run on the pre-processed data obtained from the service descriptions in the previous phase. After that, the next phase is started, as shown in Figure 8.

Here, Naive Bayes is deployed. This classifier is a member of simple probabilistic classifiers based on the Bayes' theorem. On one hand, it considers the independence assumption for each feature, which is not necessary to have a large amount of training data [48]; therefore, it could be of real benefit in most cases, especially the service domain. In addition, it seems to be a straightforward approach, an accurate multi-class predictor, and useful for the massive datasets. In terms of accuracy, the NB classifier can perform better than the regression models with the same amount of training data [49]. The algorithm allows us to predict a class, with a set of features by using the probability.

At this phase, the extracted data received from the first phase is applied as the input. These essential elements were the service name, descriptions, and IOPE features. In our example, the extracted data for the car price service is made up of: {(car price service), (find, company, offer, same, car, range, given, amount, money), (Automobile, price), (car, amount of money), (price, company)}. After performing the algorithm, this service is classified in the economic domain.

Admittedly, searching for the desired services in a class is much more comfortable than searching the whole repository. We define a function in Equation 6 that computes the service similarity to separate the unrelated services.

Afterward, the similarity computation is started, as depicted in Figure 8. The service matching

method could smoothly be performed in a specific domain.

3.4. Matchmaking Phase

Service matching is defined as finding a suitable service based on a user query meaning that the founded services should satisfy the requirements. As mentioned earlier, the same OWL ontology is considered for both sides. The matchmaking phase has two steps: class matching and service matching. Since the service matchmaking is still a time-consuming process and contains plenty of overloads when the number of services grows sharply, we designed the service classification before matchmaking. Consequently, finding the most relevant class is conducted in class matchmaking considered to reduce the matching process time; besides, the similarity value between the services and the request is determined and ranked by the equation in the service matching stage.

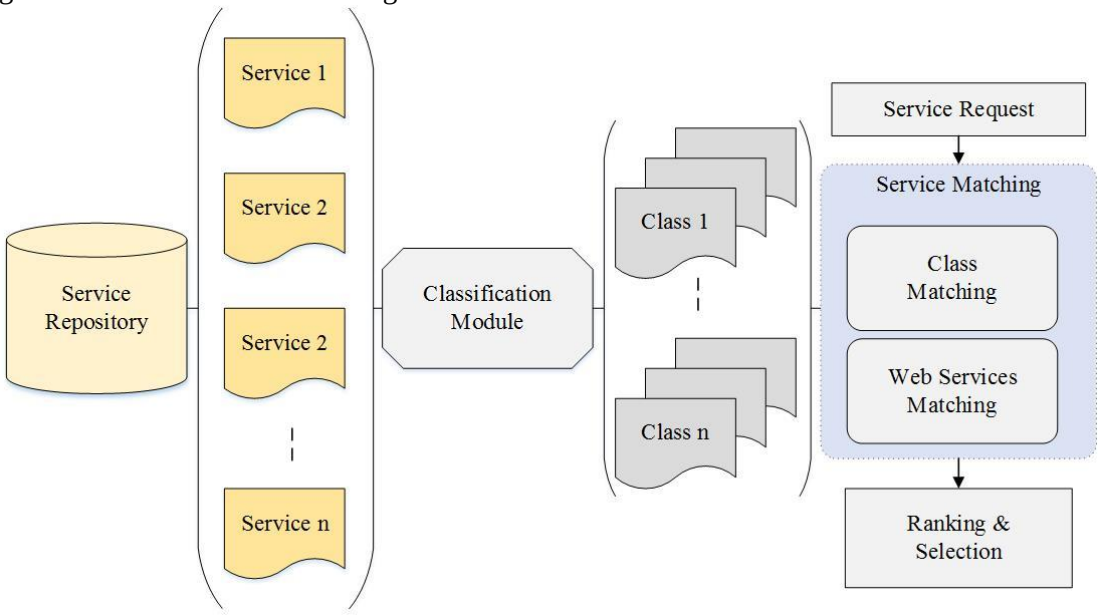

Figure 8. Classification diagram.

Class matching: After the termination of the last two phases, crystal query and classified services are employed as the inputs of this phase. The first step is the class finding or class matchmaking that is performed based on the Wikipedia LSA model. The LSA model contains a document-term matrix in which the columns and rows correspond to the terms and documents in the collection, respectively. Assuming that there are M row and N columns in the matrix, therefore, cell w*d that should satisfy the two conditions w<M and d<N shows the frequency of the word w in document d. Singular Value Decomposition (SVD) is also used by the LSA model in order to reduce the dimensions [50]. The initial size of this sparse matrix used in LSA on Wikipedia is 68,187 * 3,550,591 and the number of non-zero values is 284,093,540. As the LSA model is based on the vectors, cosine between the vectors must be computed for finding the similar degree of two concepts based on the LSA vector representations In addition, the content word lemmas are just considered in the LSA model since they can be obtained as literals in WordNet ontology [51].

The cosine similarity is the dot product between two vectors that are normalized in LSA space. If V(word) shows the vector of a word, then the cosine is calculated by Equation 2.

$$\text{LSA}(\text{word}_1, \text{word}_2) = \frac{V(\text{word}_1) * V(\text{word}_2)}{\|V(\text{word}_1)\| * \|V(\text{word}_2)\|} \quad (2)$$

The value of this equation is in the interval [0,1]. Zero signifies completely different and one indicates equivalent [52].

The light verbs, adjectives, and adverbs are excluded in this model due to three reasons, the first of which is they can not add extra meaning to the phrases. Secondly, the adjectives most of the time may act as a bridge between the nouns most of the time, and we could ignore them. Finally, the adverbs describe the state of the adjectives and verbs so they could be omitted as well. For example, ‘blue bus’ and ‘blue flower’ may be considered as two similar phrases by most similarity measurement modules due to an identical adjective. This event leads to an overestimate of the similarity measure. However, having said that, the contexts are completely different.

In addition to the Wikipedia LSA model, two similarity metrics are used in the matching process. These two metrics are selected from six popular semantic similarity measures based on their higher accuracy in paraphrase and entailment identification. The Jiang & Conrath and Leacock & Chodorow methods have a higher accuracy than the other metrics [43]. These approaches that are usually applied to calculate the similarity between concepts are path-based [29]. The Leacock & Chodorow similarity metric in [41] has been obtained from Equation 3.

$$\text{Sim}_{\text{lch}} = -\log \frac{\text{length}}{2D} \quad (3)$$

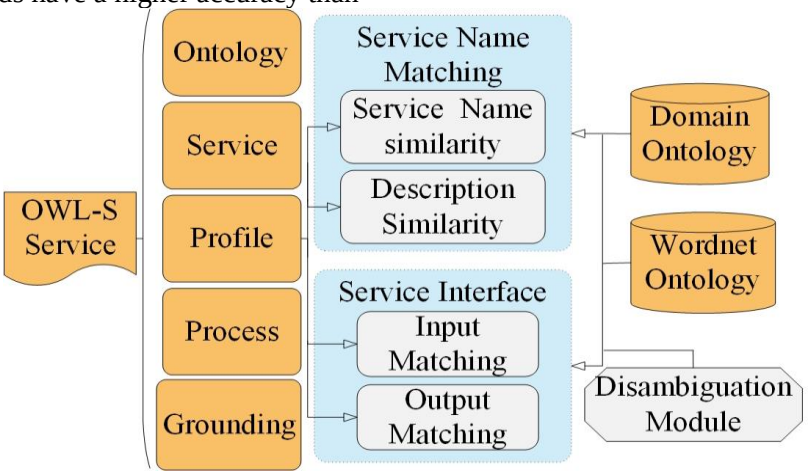

Figure 9. Web service matching (second step of matchmaking phase).

Length, which is the length of the shortest path between two words, is determined by node counting, and *D* shows the maximum depth of the taxonomy.

Jiang & Conrath also obtained the formula in Equation 4 that gives a number in the [0,1] interval [45]:

$$\text{Sim}_{\text{jcn}} = \frac{1}{\text{IC}(\text{concept 1}) + \text{IC}(\text{concept 2}) - 2 * \text{IC}(\text{LCS})} \quad (4)$$

where *LCS* is the short form of the least common subsumer and *IC* is Information Content described as equation below:

$$\text{IC}(c) = -\log P(c) \quad (5)$$

where P(c) determines the probability of occurrence of a sample of concept c of a sizeable dataset. On the whole, we propose a concept similarity by the following equation:

$$\textit{Concept Similarity} = \sqrt[3]{\text{Sim}(\text{lch}) * \text{Sim}(\text{jcn}) * \text{LSA}(w1, w2)} \quad (6)$$

First, each word of the crystal query is compared to the classes' name based on Equation 6. Then the class with higher concept similarity is selected for the web service matching step.

For our example, the LSA values are LSA(car, domains) and LSA(price, domains) in the car price request, where the domains are defined as the classes' name, and they are specified as follow: economy, food, education, weapon, communication, travel, and medical. Finally, the "economy" is selected as the most relevant domain for web service matching because its concept similarity value is higher than the other ones. Furthermore, that Sim(lch) and Sim(jcn) are calculated based on two cases given in Table 3.

Table 3. Cases used in sim module.

Cases	Concept 1	Concept 2
Case 1	Car	Communication Economy education
Case 2	Price	food medical travel weapon

Web service matching: After crossing the class matching, a two sub-step process constitutes this step, during which service name matching proceeds the second sub-step, which is service interface matching.

The functional similarity is calculated between the service request and the service file. It is defined as a geometric mean composed of service similarity (service name, service description), and service interface similarity (input and output) in Equation 7.

$$\text{Semantic Similarity Measure} = \sqrt[2]{\text{ServiceSimilarity} * \text{ServiceInterfaceSim}} \quad (7)$$

Equation 7 is used to calculate the overall semantic similarity. ServiceSimilarity and ServiceInterfaceSim are obtained from the related algorithms, which are described in the above section formerly.

In Figure 8, the service profile is used for web service matching, which contains service name, text description, input, output, and process. This scheme describes the modular-based process of service matching, considering semantic similarity. Initially, the match engine calculates the service name matching between the user request and each service. Likewise, an identical process is carried out for the input and output of each service based on the Service Interface Matching module. The service name similarity, description similarity, and interface similarity between each service and the request are calculated by the ServiceInterfaceSim and ServiceSimilarity algorithms shown in Figure 3. As it can be seen from Figure 9, a disambiguation module is required for word sense disambiguation (WSD) using ontologies for enhancing the accuracy of the proposed method.

In the case of our example, 38 services are identified in the economy domain, and they are related to car price (the similarity threshold is 0.75). After class matching, the most relevant services are sorted, and they are listed in Table 4.

Table 4. most relevant services found by method for the given example

Service names	Inputs	Outputs
Car price service	Car	Price
Car Year Price Service	Car	Price Year
Car price service	Car	Price TaxedPrice
Car Price quality	Car	Price Quality
Car price service	Car	Auto Price
Car price report service	Car	Price Report

As mentioned earlier, the semantic similarity measure is calculated for the above example (car price request) based on Equation 7, and the most relevant services are retrieved.

This work focuses on the discovery process. The next step is the selection part in the web service life cycle, which includes the criteria required for choosing one or some of the most suitable services retrieved from the discovery process. An example of such criteria is the QOS-based selection metrics published in [53]. In that paper, we focused on the metrics and criteria required for service selection according to the quality of services parameter with the ontology scheme.

4. Evaluation of Web Service Discovery

As mentioned earlier, the OWL-S test collection, version 3, with 1007 instances was deployed to evaluate the proposed method. This dataset was prepared by a semantic web central group. It is important to note that no standard dataset for the OWL service retrieval does exist yet. As a result, OWLS-TC can only be selected as one potential collection for related research as long as a standard dataset is designed. The archive of the OWL-S test collection comprises the following subdirectories:

- Services: consist of all of the test collection that is divided into seven domains (education, medical care, food, travel, communication, economy, and weapon).
- Queries: comprise all the services requests of the test collection.
- Ontology: is made up of the ontologies used by the services, and the service requests are presented in this folder.
- Domains: consist of all the services sorted according to the domains.
- WSDL: consists of the WSDL groundings for all of the services.

After the pre-processing step, an extraction function was designed to save all the main requirements of each web service description in a text file. Next, the functional similarity was computed using Equation 7. Then the web services were grouped into seven classes by applying the Naive Bayes text classification algorithm, by which each class showed a specific domain. These domains were defined the same as the OWL-S test collection.

The robustness of the classifier was tested in two ways. First, it was tested by intentionally making slight changes to the training data and adding noisy data randomly to the collection such as the unrelated texts and numbers; nevertheless, these perturbations had a minimal impact on our results. Secondly, after the classification process, our classes were compared with the OWL-S TC classes with respect to the number of the services and their names, which meant that we had checked all our results versus the TC services according to seven categories. Table 5 illustrates examples of the services extracted from OWL-S TC. Moreover, an example of how web services are matched based on our scheme is presented in the proposed method section.

The classification step was executed based on the specific computer with the following configuration:

- Operating System: Ubuntu-Linux
- CPU: Intel dual-core (2.5 GHz)
- Memory: 4 GB

– Execution time: less than 25 seconds

Two models were employed as the reference models for verifying the correctness of our retrieved results for each request. Indeed, they determined whether the discovered web services by the model were correctly found for each query or not.

The first model was proposed by M Klusch *et al*. [54]. Their classification can be used as a suitable benchmark to evaluate the results. Furthermore, Chen *et al*. [29] requested 161 the undergraduate students to determine and classify the services with the help of a query set. We considered both of them to compare our results.

One more thing, a similarity threshold is required to determine whether a service corresponds to a query relying on the proposed similarity approach and policy. Therefore, we used 0.75 as the similarity threshold since the services should be arranged as either a mismatch or match.

The discovery methods' evaluation is categorized in the domain of information retrieval, where three parameters are generally applied to measure relatedness. The first parameter, precision (positive predictive value), shows how many selected samples are relevant. The other metric is the recall (sensitivity), which means how many relevant samples are selected [29]. Finally, the F-score is determined from the harmonic mean of the Precision and Recall measures. Indeed, F-score intensifies the impact of small outliers and alleviates the impact of big ones [29].

The proposed method is compared with three other methods-a lexical matching approach and two semantic similarity-based methods. Lexical matchmaking, considered the basic one, and is known to do well for the web services [36]. Besides, the second method is one of the newest research works in the related field [29]. The first technique was applied to calculate the cosine degree between keyword–document vectors using their textual descriptions. After the term tokenization, the term frequency-inverse document frequency[1] was applied to weight the terms extracted from the textual description of services. The F-score reported by their model is nearly 0.7. Therein lies the problem as to why it acted poorly; lexical matching approaches suffer from the lack of true meaning of the words. By the application of semantic and the usage of the clustering method, Fuzan Chen *et al*. [29] performed the discovery procedure. Furthermore, they applied concept similarity considering oppositeness along with the process and functional similarity. All in all, they could succeed in gaining 91% in terms of F-score. In more detail, the methods that just rely on syntactic matching are not able to achieve a higher accuracy than the ones that take into account the semantic meaning.

Another method proposed by Pushpa *et al*.[31] was implemented through four steps, two important of which are the generation of synonyms using the thesaurus and semantic extraction. A word-level semantics is performed along with clustering and NPMI value. They applied semantic clustering by a thesaurus and interface mining. Then web service ontology was added by index library. All these steps are done to prepare data for onto structured index. In comparison with our pre-processing pipeline, these steps are more complex. Finally, they are delivered to the discovery phase, which, in turn, results in 90% of the F-score.

However, the above method reached a greater depth of matching degree; they applied a more computationally expensive procedure to accomplish the desired task due to including all aspects of the published services along with a graph-based method to reduce the error significantly. As opposed to the aforementioned approaches, we rely on the semantic disambiguation method by applying a two-tier related word sense, which not only increases the depth of understanding descriptions for both sides but also helps to accomplish the matching process to a better extent in terms of metrics, as follows.

Five information retrieval metrics typically applied in all the similar research works are employed to calculate the performance of our system. As it can be viewed in Table 6, the error rate was reduced; also, the false discovery rate, sensitivity, positive predictive value, miss rate, and F1 score were calculated.

As mentioned earlier, we used the same similarity threshold as [29], indicating the best point for similarity threshold that was 0.75. The precision and recall were obtained to be 0.96 and 0.94, respectively. Altogether, our results were better than the keyword-based method (recall = 0.68, precision = 0.7), and the semantic method's precision of 0.93 and recall of 0.90. These statistics are listed in Table 7. Different similarity thresholds held almost the same result. The results obtained demonstrate that the combination of a correct classifier alongside an optimal matching algorithm makes a significant contribution to the service discovery.

[1] TF-IDF

Table 5. Examples of our web services.

Web service	Service name	Service description	inputs	outputs
Web service 1	GetDrugInformation	Delivers the required drug names for a treatment	TreatmentIInformation	RequiredDrugs
Web service 2	CountryWeatherFront Service	This service informs about weatherfront in a given country	COUNTRY	WEATHERFRONT
Web service 3	HEBFoodService	This service returns food of HEB grocery company	FOOD	FOOD_PROCESS

Five information retrieval metrics typically applied in all the similar research works are employed to calculate the performance of our system. As it can be viewed in Table 6, the error rate was reduced; also, the false discovery rate, sensitivity, positive predictive value, miss rate, and F1 score were calculated. As mentioned earlier, we used the same similarity threshold as [29], indicating the best point for similarity threshold that was 0.75. The precision and recall were obtained to be 0.96 and 0.94, respectively. Altogether, our results were better than the keyword-based method (recall = 0.68, precision = 0.7), and the semantic method's precision of 0.93 and recall of 0.90. These statistics are listed in Table 7. Different similarity thresholds held almost the same result. The results obtained demonstrate that the combination of a correct classifier alongside an optimal matching algorithm makes a significant contribution to the service discovery. It can be seen from Table 6, that the error rate is reduced. The false discovery rate, sensitivity, positive predictive value, miss rate, and F1 score were calculated based on Table 6. The results for the two other techniques are not shown and compared in Table 6 because their confusion matrices are not available. These metrics are the most common ways in the performance evaluation of the information retrieval systems (IR systems).

Table 6. Exprimental measurement.

Performance metric of IR system	Formula	Value
False discovery rate (FDR)	$\mathrm{FDR} = \frac{FP}{FP + TP}$	0.03
Sensitivity or true positive rate (TPR)	$\mathrm{TPR} = \frac{TP}{TP + FN}$	0.95
precision or positive predictive value (PPV)	$\mathrm{PPV} = \frac{TP}{TP + FP}$	0.97
miss rate or false-negative rate (FNR)	$\mathrm{FNR} = \frac{FN}{FN + TP}$	0.049
F1 score	$\mathrm{F1} = \frac{TP}{2TP + FP + FN}$	0.48

In general, by inspecting Table 1 and the results of Table 7, some points can be grasped. First, both the OntoDisco and keyword-based methods used interface as the matching object but the former significantly indicate a better precision and recall. Therefore, it can be inferred that the matching objects are not so influential that we can yield better results. Moreover, both the semantic method and keyword-based method made use of the same public ontology but the former one could act far better than the latter nonetheless. The reason this occurred lies at the root of the usage of the semantic model to boost perception for the matching process. It is clear that all the methods (except for keyword-based), which have the usage of semantic approach in common, lead to better results than the traditional methods.

5. Conclusion and Future Works

A web service discovery method primarily based on the semantic matching and classification phases was designed to achieve an effective discovery method, which not only could reduce the error rate of the previously mentioned methods but also was suitable to extend and evaluate on a larger dataset. On one hand, the classification procedure assigns services to a suitable domain, which means that it can narrow the search area and enable a fast semantic matching in huge pools, consequently. On the other hand, the proposed discovery mechanism ensures a more accurate semantic matching and the velocity of the operations. Our experiments confirmed that a mixture of the Wikipedia LSA model with semantic similarity metrics worked best for the matchmaking process. We also found that the word sense disambiguation could compensate the user query restrictions. In summary, it could be concluded that the precision error rate was reduced by 43% in the proposed method, and the recall error rate was decreased by 40%.

Table 7. Comparison results of methods.

Methods	Precision	Recall	F-Score
Proposed method	0.96	0.94	0.949
Keyword-based method	0.7	0.68	0.689
Semantic method [29]	0.93	0.9	0.914
OntoDisco	0.91(max)	0.89(max)	0.90

However, this work had some limitations. During the discovery process, if updating the services was dynamic, and then the classification phase would be a time-consuming process for assigning new

objects (services). Moreover, the restricted data collection was a dominating factor that limited the assessment of the generalizability of our model. In the future, we wish to develop our model by importing a non-functional constraint to the problem for a further investigation in service discovery. In addition, it would be interesting to know how our method performs on texts of larger length and big data problems.

References

[1] F. Devin, Web oriented architecture–How to design a RESTFull API. TORUS 1–Toward an Open Resource using Services: Cloud Computing for Environmental Data, pp.191-206, 2020.

[2] S. Adeli, and P. Moradi, QoS-based Web Service Recommendation using Popular-dependent Collaborative Filtering. Journal of AI and Data Mining, 8(1), pp.83-93, 2020.

[3] M. Klusch, P. Kapahnke, S. Schulte, F. Lecue, and A. Bernstein, "Semantic Web Service Search: A Brief Survey," *KI-Künstl. Intell.*, pp. 1–9, 2015.

[4] K.-H. Lee, M. Lee, Y.-Y. Hwang, and K.-C. Lee, "A framework for xml web services retrieval with ranking," in *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, 2007, pp. 773–778.

[5] N. E. Evangelopoulos, "Latent semantic analysis," *Wiley Interdiscip. Rev. Cogn. Sci.*, Vol. 4, No. 6, pp. 683–692, 2013.

[6] T. K. Landauer, "LSA as a theory of meaning," in *Handbook of latent semantic analysis*, Psychology Press, 2007, pp. 15–46.

[7] I. Lizarralde, C. Mateos, A. Zunino, T. A. Majchrzak, and T. M. Grønli, Discovering web services in social web service repositories using deep variational autoencoders. Information Processing and Management, 57(4), p.102231, 2020.

[8] M. Fariss, N. El Allali, H. Asaidi, and M. Bellouki, October. Review of ontology based approaches for web service discovery. In International Conference on Advanced Information Technology, Services and Systems (pp. 78-87). Springer, Cham, 2018.

[9] O. Sharifi, Z. Bayram, A critical evaluation of web service modeling ontology and web service modeling language. In International Symposium on Computer and Information Sciences (pp. 97-105). Springer, Cham, 2016.

[10] C. Mateos, M. Crasso, A. Zunino, and M. Campo, "Supporting ontology-based semantic matching of web services in movilog," in *Advances in Artificial Intelligence-IBERAMIA-SBIA 2006*, Springer, 2006, pp. 390–399.

[11] M. Shamsfard and A. A. Barforoush, "Learning ontologies from natural language texts," *Int. J. Hum.-Comput. Stud.*, Vol. 60, No. 1, pp. 17–63, 2004.

[12] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, Vol. 18, No. 11, pp. 613–620, 1975.

[13] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic matching of web services capabilities," in *International Semantic Web Conference*, 2002, pp. 333–347.

[14] W. Jiang, J. Lin, H. Wang, and S. Zou, Hybrid semantic service matchmaking method based on a random forest. Tsinghua Science and Technology, 25(6), pp.798-812, 2020.

[15] S. Pakari, E. Kheirkhah, and M. Jalali, "A Novel Approach: A Hybrid Semantic Matchmaker For Service Discovery In Service Oriented Architecture," *Int. J. Netw. Secur. Its Appl.*, Vol. 6, No. 1, p. 37, 2014.

[16] V. Oleshchuk, "Ontology-based service matching and discovery," in *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on*, 2011, Vol. 2, pp. 609–612.

[17] A. V. Paliwal, B. Shafiq, J. Vaidya, H. Xiong, and N. Adam, "Semantics-based automated service discovery," *Serv. Comput. IEEE Trans. On*, Vol. 5, No. 2, pp. 260–275, 2012.

[18] Y. Shi, G. Li, J. Li, and Y. Li, "Framework of semantic web service discovery based on ontology mapping," in *Research Challenges in Computer Science, 2009. ICRCCS'09. International Conference on*, 2009, pp. 77–80.

[19] S. Pakari, E. Kheirkhah, and M. Jalali, "Web service discovery methods and techniques: A review," *Int. J. Comput. Sci. Eng. Inf. Technol.*, Vol. 4, No. 1, 2014.

[20] B. Di Martino, "Semantic web services discovery based on structural ontology matching," *Int. J. Web Grid Serv.*, Vol. 5, No. 1, pp. 46–65, 2009.

[21] L. Zhou, "An approach of semantic web service discovery," in *Communications and Mobile Computing (CMC), 2010 International Conference on*, 2010, Vol. 1, pp. 537–540.

[22] A. B. Bener, V. Ozadali, and E. S. Ilhan, "Semantic matchmaker with precondition and effect matching using SWRL," *Expert Syst. Appl.*, Vol. 36, No. 5, pp. 9371–9377, 2009.

[23] C. Ke and Z. Huang, "Self-adaptive semantic web service matching method," *Knowl.-Based Syst.*, Vol. 35, pp. 41–48, 2012.

[24] A. Adala, N. Tabbane, and S. Tabbane, "A framework for automatic web service discovery based on semantics and NLP techniques," *Adv. Multimed.*, Vol. 2011, p. 1, 2011.

[25] M. D. Lakshmi and J. P. M. Dhas, "A Hybrid Approach for Discovery of OWL-S Services Based on Functional and Non-Functional Properties," *WSEAS Trans Comput*, Vol. 14, pp. 62–71, 2015.

[26] R. Karimpour and F. Taghiyareh, "conceptual discovery of web services using WordNet," in *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*, 2009, pp. 440–444.

[27] G. Ganapathy and C. Surianarayanan, "An approach to identify candidate services for semantic web service discovery," in *2010 IEEE international conference on service-oriented computing and applications (SOCA)*, 2010, pp. 1–4.

[28] Y. Peng and C. Wu, "Automatic semantic web service discovery based on assignment algorithm," 2010.

[29] F. Chen, M. Li, H. Wu, and L. Xie, "Web service discovery among large service pools utilising semantic similarity and clustering," *Enterp. Inf. Syst.*, Vol. 11, No. 3, pp. 452–469, 2017.

[30] D. Lin and others, "An information-theoretic definition of similarity.," in *Icml*, 1998, Vol. 98, pp. 296–304.

[31] C. N. Pushpa, G. Deepak, A. Kumar, J. Thriveni, and K. R. Venugopal, "OntoDisco: improving web service discovery by hybridization of ontology focused concept clustering and interface semantics," in *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pp. 1–5, 2020.

[32] C. B. Merla, "Context-aware match-making in semantic web service discovery," *IJAEST-Int. J. Adv. Eng. Sci. Technol.*, Vol. 1, No. 9, pp. 243–247, 2010.

[33] M. Klusch, P. Kapahnke, and B. Fries, "Hybrid semantic web service retrieval: A case study with OWLS-MX," in *Semantic Computing, 2008 IEEE International Conference on*, 2008, pp. 323–330.

[34] J. R. Raj and T. Sasipraba, "web service discovery based on computation of semantic similarity distance and Qos normalization," *Indian J. Comput. Sci. Eng.*, Vol. 3, No. 2, pp. 235–239, 2012.

[35] A. Farooq and R. Arshad, "An Efficient Technique for Web Services Identification," *Int J Multidiscip Sci Eng*, Vol. 2, No. 1, pp. 26–30, 2011.

[36] N. Kokash, "A comparison of web service interface similarity measures," *Front. Artif. Intell. Appl.*, Vol. 142, p. 220, 2006.

[37] "SemWebCentral: OWL-S Service Retrieval Test Collection: File Release Notes and Changelog." http://projects.semwebcentral.org/frs/shownotes.php?release_id=369.

[38] "Porter-Stemmer," *Drupal.org*. https://www.drupal.org/project/porterstemmer.

[39] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit.," in *ACL (System Demonstrations)*, 2014, pp. 55–60.

[40] K. Clark and C. D. Manning, "Improving coreference resolution by learning entity-level distributed representations," *ArXiv Prepr. ArXiv160601323*, 2016.

[41] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification," *WordNet Electron. Lex. Database*, Vol. 49, No. 2, pp. 265–283, 1998.

[42] X. Zhu, X. Yang, Y. Huang, Q. Guo, and B. Zhang, Measuring similarity and relatedness using multiple semantic relations in WordNet. Knowledge and Information Systems, 62(4), 1539-1569, 2020.

[43] C. Corley and R. Mihalcea, "Measuring the semantic similarity of texts," in *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, 2005, pp. 13–18.

[44] M. C. Lintean and V. Rus, "Measuring Semantic Similarity in Short Texts through Greedy Pairing and Word Semantics.," 2012.

[45] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *ArXiv Prepr. Cmp-Lg9709008*, 1997.

[46] Y. Peng, "Two levels semantic web service discovery," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*, 2010, Vol. 6, pp. 2523–2526.

[47] M. Lintean and V. Rus, "An Optimal Quadratic Approach to Monolingual Paraphrase Alignment," in *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*, 2015, pp. 127–134.

[48] S. Xu, Bayesian Naïve Bayes classifiers to text classification. Journal of Information Science, 44(1), pp. 48-59, 2018.

[49] B. T. Pham, D. Bui, I. Prakash, and M. Dholakia, "Evaluation of predictive ability of support vector machines and naive Bayes trees methods for spatial prediction of landslides in Uttarakhand state (India) using GIS," *J Geomat.*, Vol. 10, pp. 71–79, 2016.

[50] D. Ștefănescu, R. Banjade, and V. Rus, "Latent semantic analysis models on wikipedia and tasa," 2014.

[51] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Process.*, Vol. 25, No. 2–3, pp. 259–284, 1998.

[52] M. C. Lintean, C. Moldovan, V. Rus, and D. S. McNamara, "The Role of Local and Global Weighting in Assessing the Semantic Similarity of Texts Using Latent Semantic Analysis.," in *FLAIRS Conference*, 2010, pp. 235–240.

[53] P. Farzi, R. Akbari, and O. Bushehrian, "Improving semantic web service discovery method based on QoS ontology," in *2017 2nd conference on swarm intelligence and evolutionary computation (csiec)*, 2017, pp. 72–76.

[54] M. Klusch, B. Fries, and K. Sycara, "OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services," *Web Semant. Sci. Serv. Agents World Wide Web*, Vol. 7, No. 2, pp. 121–133, 2009.

یک روش ترکیبی موثر برای شناسایی معنایی سرویس های وب

پوریا فرضی و رضا اکبری*

دنشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی شیراز، شیراز، ایران.

ارسال ۲۰۲۰/۰۸/۱۰؛ بازنگری ۲۰۲۱/۰۶/۲۵؛ پذیرش ۲۰۲۱/۱۰/۰۷

چکیده:

سرویس وب یک فناوری برای تعریف اشیاء خود توصیف کننده و برنامه‌های کاربردی مبتنی بر ساختار و اتصال ضعیف است. آنها در سراسر وب قابل دسترسی هستند و یک بستر انعطاف پذیر را ارائه می‌دهند. اگرچه رجیستری های سرویس مانند توصیف جهانی، شناسایی و یکپارچه سازی (UDDI) امکاناتی را در اختیار کاربران قرار می‌دهد تا بتوانند نیازمندیها را جستجو کنند، اما بازیابی نتایج دقیق که نیازهای کاربران را برآورده می‌کند، همچنان یک کار دشوار است، زیرا ارائه دهندگان و درخواست کنندگان خدمات دیدهای متفاوتی در مورد توصیف سرویس دارند. در نتیجه، یکی از چالش برانگیزترین موانع موجود در کار شناسایی سرویس، نحوه درک هر دو طرف خواهد بود، که به آن فهم مبتنی بر دانش می‌گویند. این برای موتورهای جستجو، وظایف بازیابی اطلاعات و حتی کارهای مختلف مبتنی بر پردازش زبان طبیعی ارزش فوق العاده ای دارد. هدف این است که به تشخیص دقیق درجه‌های تطبیق و بازیابی مناسب ترین سرویس به طور مستقیم کمک کنیم. در این کار تحقیقاتی، ما یک روش تشابه مفهومی را به عنوان روشی شناسایی سرویس معرفی میکنیم که وابستگی کمتری به توصیفهای ارائه دهنده و توضیحات کاربر نیاز دارد تا مداخله دستی را برای هر دو طرف کاهش دهد و اطلاعات واضح تری را برای ماشین‌ها ارائه دهد. ما با استفاده از مدل تجزیه و تحلیل معنایی نهفته (LSA) در طرح هستی شناسی-WordNet و الگوریتم شباهت مبتنی بر زمینه خاص، یک رویکرد جامع مبتنی بر دانش ارائه می‌دهیم. ارزیابی روش شباهت ما، که بر روی مجموعه آزمون OWL-S انجام شده است، نشان می‌دهد که یک الگوریتم شباهت حس-زمینه می‌تواند روش ابهام زدایی توصیفات را افزایش دهد، که منجر به وضوح مفهومی می‌شود. روش پیشنهادی عملکرد کشف خدمات را در مقایسه با روشهای جدید مبتنی بر کلمات کلیدی و مبتنی بر معنایی بهبود می بخشد.

کلمات کلیدی: وب سرویس معنایی، انطباق، دسته بندی، تحلیل معنایی نهفته، معنا شناسی کلمه.